UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/825,694	04/16/2004	John Harper	P3353US1 (119-0041US)	9161

61947 7590 04/09/2009
WONG, CABELLO, LUTSCH, RUTHERFORD & BRUCCULERI LLP
20333 SH 249
SUITE 600
HOUSTON, TX 77070

EXAMINER	
GUERTIN, AARON M	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
04/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/825,694	Applicant(s) HARPER ET AL.	
	Examiner AARON M. GUERTIN	Art Unit 2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 January 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *8-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *8-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *16 June 2008* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

- Claims 8-39 are presented for examination.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 8-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Publication No.: US 2003/0011637 A1 (Boudier), in view of U.S. Publication No.: US 2002/0109682 (Nash) in further view of U.S. Publication No.: US 2005/0041031 A1 (Diard).

3. Regarding claim 8, Boudier teaches of a method of creating an image graph ([0092] - *In this optimization, the textures used in a scene can be saved in the scene graph file itself, instead of an external image file. It may be desirable to scale down the size of all the textures of a given scene graph file, so that there is no need (or a reduced need) to page textures. The resizing of the image can be done based on the size of the texel...*), said image graph comprising one or more nodes ([Fig. 9, (910, 920, 950)]), inputs to those nodes, and outputs from those nodes (each node of Fig. 9 (900) is linked; also see [0054]-[0058]), the method comprising the steps of: the method

comprising the steps of: **optimizing said image graph by running software on a CPU; compiling said image graph by running software on said CPU**; (processor 504 is regarded as the CPU; the compiling is the optimization of the graph by executing the program(s) [0007] - *The method of the invention includes the steps of receiving an input scene graph, creating the optimization process, applying the optimization process to the input scene graph, and post-optimization processing…* [0032] - *The set of available atomic optimizations is contained in an optimization base 425. A list of the available atomic optimizations is maintained in an optimization registry 430, along with information pertinent to the execution of the specific atomic optimizations. This information can include, for example, the parameters required by an atomic optimization, and any priority information that defines the sequence in which specific atomic optimizations can or should be applied…* [0037] - *Referring again to FIG. 3, optimizer 330 may be implemented using hardware, software or a combination thereof. In particular, optimizer 330 may be implemented using an object-oriented approach, and execute on a computer system or other processing system. An example of such a computer system 500 is shown in FIG. 5. The computer system 500 includes one or more processors, such as processor 504…*); and **rendering said image graph by running said compiled image graph** ([0029] - *The system of the invention optimizes an input scene graph to produce an optimized scene graph which can then be rendered. The context in which the optimizer functions is illustrated in FIG. 3. A modeler 310a creates a scene graph 315a and passes it to a common export library 320…*).

Boudier teaches the limitations of claim 8 above, however Boudier fails to specifically teach of **wherein the node(s) are program(s) and wherein executing the scene graph yielding a rendered image**.

Nash is analogous art that further teaches of **wherein the node(s) are program(s)** ([Fig. 3] and [0054]; furthermore [Fig. 6] shows phase modules (nodes) of processing and each node comprises a specific program code to conduct the function within said node in a sequence) **and wherein executing the modules yields a rendered image** ([0060], [0061] and [0062] - *The phase modules, containing microinstructions for a phase or sub-phase, are stored in phase code depository 122… The phase modules (excluding the output phase module) are interchangeable and capable of being synthesized in any order to implement a desired mode… Typically in computer graphics, the user supplies function calls to set the vertices, normals, primitives, textual coordinates, other operations, or the like to render an image…).*

All the elements of claim 8 are known in Boudier in view of Nash, the only difference is the combination of known elements into a single system and method.

Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to include wherein the nodes of Boudier are modules indicating a function that needs to be processed and wherein the execution of the modules renders an image in Boudier, as doing so would provide the means for effectively optimizing the calculations of the attributes to a specific image instead of only optimizing the steps of applying the attributes, reducing overall system calculations and increases system efficiency and bandwidth.

Boudier and Nash teach the limitations of claim 8 above; however, Boudier and Nash fail to specifically teach of wherein the **programs can be run on a GPU** (GPU programs) and therefore the compilation is conducted **on a GPU.**

Diard is analogous art that further teaches of wherein the **programs can be run on a GPU** (GPU programs) and therefore the compilation is conducted **on a GPU** ([0033] - *In operation, a graphics driver program (or other program) executing on CPU 102 delivers rendering commands and associated data for processing by GPUs 114a, 114b… a rendering command may be associated with rendering data, with the rendering command defining a set of rendering operations to be performed by the GPU on the associated rendering data. In some embodiments, the rendering data is stored in the command buffer adjacent to the associated rendering command.*).

All the elements of claim 8 are known in Boudier and Nash in view of Diard, the only difference is the combination of known elements into a single system and method.

Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to include implementating the processing the compilation on a GPU in Boudier, as doing so would provide the means and advantages of a higher throughput optimized, faster processing capabilities, and more advantageous memory bandwidth.

4. Regarding claim 9, Boudier, Nash, and Diard teach the limitations of claim 8 above, the rationale disclosed in the rejection incorporated herein and Boudier further teaches of **wherein the step of optimizing includes the step of using a cache look-up to see if said rendered image is already in cache** ([0094] - *When creating*

attributes in a scene graph, it is possible to have multiple attributes that represent the same state all sharing the same set of objects. A created texture state object, for example, can be used at multiple points in a scene graph. This optimization makes sure that any two attributes that represent the same state are sharing the same objects. This sharing of objects improves memory usage, and can also improve run time efficiency, since a cache can use a pointer to the object as an identifier.).

5. Regarding claim 10, Boudier, Nash and Diard teach the limitations of claim 8 above, the rationale disclosed in the rejection incorporated herein and Boudier further teaches **wherein the step of optimizing includes the step of using a cache look-up to see if said image graph has already been optimized and is in a memory** (at step 630 the create optimization process executes and actually begins on step 710 wherein the memory is checked for a former optimized scene graph wherein upon extra parameters may be added if different from the existent optimization; and [Fig. 6] and [Fig. 7]; [0032] - *The set of available atomic optimizations is contained in an optimization base 425. A list of the available atomic optimizations is maintained in an optimization registry 430, along with information pertinent to the execution of the specific atomic optimizations. This information can include, for example, the parameters required by an atomic optimization, and any priority information that defines the sequence in which specific atomic optimizations can or should be applied. Given the choice of a specific atomic optimization identified in user configuration information 415, optimization manager 410 associates input scene graph 405 with the identified atomic optimization in*

optimization base 425, via optimization registry 430… [0046] - *Step 630 above, the creation of an optimization process, is illustrated in greater detail in FIG. 7. This process begins at step 705. In step 710, the optimization manager receives input from a user regarding a specific atomic optimization to be used, along with any parameters that the user elects to specify.*).

6. Regarding claim 11, Boudier, Nash and Diard teach the limitations of claim 8 above, the rationale disclosed in the rejection incorporated herein and Boudier further teaches **wherein the step of optimizing includes the step of calculating an intersection, said intersection representing an area where said rendered image is both defined by said image graph and part of a region requested by a process running on said CPU that has requested creation of said image** (NOTE: according to applicants specification [0089] - Core Imaging performs node reduction analysis and eliminates nodes where possible. After unnecessary (or collapsible) nodes are optimized, Core Imaging moves to step 7103 where optimization is performed to ultimately limit the size of buffers and image inputs. This step involves intersecting two regions called domain of definition ("DOD") and region of interest ("ROI`). After the ROI/DOD optimization, the graph is ready to compile in step 7104… and [0111] - In developing optimization techniques, the domain of definition ("DOD") is interesting because there is no need to compute or draw pixels outside the DOD. Therefore, in optimizing a graph, there is use in first calculating the DOD of the root node (the very highest node, for example node 415 of FIG. 4).). From the disclosure in the specification

and corresponding Fig. 4 and Fig. 7 for support of claim 11, it can be seen that the intersection is where two nodes come together as the optimization performs collapsing wherein each node then becomes a DOD and the result of the two nodes that intersect and come together at another node is a ROI.

Boudier optimizes in several ways encompassing the same method of the applicant. For example, as supported by Boudier in [0053] - [0059] are methods of "collapse geometers" and Collapse hierarchy which as supported by [Fig. 9] and [Fig. 10 and 11] optimizations are performed on DODs forming new DODs (if collapsing to the top of the scene graph then obtaining an ROI). ([Figs. 9-11] and [0054] - *With this optimization, the geometry nodes are gathered and collectively replaced by geometry 980 in the optimized scene graph. In the collapse geometry scene graph optimization, the user can specify a common format for the vertex array of the resulting geometry. This format specification represents a user input for the optimization.*) Furthermore as disclosed and incorporated by claim 8, the method of optimizing within intersections and regions is conducted on a processor (CPU).

7. Regarding claim 12, it is similar in scope to claim 8 and is rejected under the same rationale.

8. Regarding claim 13, it is similar in scope to claim 8 and is rejected under the same rationale.

9. Regarding claim 14, Boudier, Nash and Diard teach the limitations of claims 8 and 11 above, the rationale disclosed in the rejection incorporated herein and Boudier further teaches the step of, **using said calculated intersection to limit the number of pixels that require calculation during said rendering** (it is implied that an optimization by combining (collapsing) DODs that the scene graph no longer has to process each of the nodes, furthermore [0055] - *Because the number of nodes is reduced, memory usage is reduced. Also, if all the geometries are to be drawn, drawing time will be reduced because there will be fewer function calls…* and [Figs. 9-11] and [0054] - *With this optimization, the geometry nodes are gathered and collectively replaced by geometry 980 in the optimized scene graph. In the collapse geometry scene graph optimization, the user can specify a common format for the vertex array of the resulting geometry. This format specification represents a user input for the optimization.*); and Diard further teaches as incorporated by claim 1 the rendering **on a GPU**.

10. Regarding claim 15, it is similar in scope to claim 14 and is rejected under the same rationale.

11. Regarding claim 16, it is similar in scope to claim 14 and is rejected under the same rationale.

12. Regarding claim 17, Boudier, Nash and Diard teach the limitations of claims 8 and 11 above, the rationale disclosed in the rejection incorporated herein and Boudier

further teaches the step of, **using said calculated intersection to limit the amount of memory necessary for storing said rendered image** ([0059] - *In general, the collapse hierarchy optimization has the benefit of enhancing the traversal time of a scene graph, and also reduces the memory usage associated with the scene graph by reducing the number of nodes.*).

13. Regarding claim 18, it is similar in scope to claim 17 and is rejected under the same rationale.

14. Regarding claim 19, it is similar in scope to claim 17 and is rejected under the same rationale.

15. Regarding claim 20, Boudier, Nash and Diard teach the limitations of claim 8 above, the rationale disclosed in the rejection incorporated herein and Boudier further teaches the step of wherein said step of optimizing comprises **the additional steps of using a cache to determine if said rendered image is available in memory** ([0094] as imported, the rationale disclosed in the rejection of claim 9); **using the CPU to perform ROI/DOD intersections with respect to one or more of said nodes** (Boudier optimizes in several ways encompassing the same method of the applicant. For example, as supported by Boudier in [0053] - [0059] are methods of "collapse geometers" and Collapse hierarchy which as supported by [Fig. 9] and [Fig. 10 and 11] optimizations are performed on DODs forming new DODs (if collapsing to the top of the scene graph then obtaining an ROI). ([Figs. 9-11] and [0054] - *With this optimization,*

the geometry nodes are gathered and collectively replaced by geometry 980 in the optimized scene graph. In the collapse geometry scene graph optimization, the user can specify a common format for the vertex array of the resulting geometry. This format specification represents a user input for the optimization.) Furthermore as disclosed and incorporated by claim 8, the method of optimizing within intersections and regions is conducted on a processor (CPU); **using the CPU to determine if nodes may be combined to form a node that has been created by combining two other nodes.** (Boudier by examples of [Fig. 12], [Fig. 13], and [Fig. 14] clearly show wherein the optimization programs (ability to collapse) may be calculated to combine). Nash further teaches that the nodes are programs ([Figs. 3 and 6] and [0060] through [0062]).

Diard further teaches of **using a CPU to determine if said GPU is capable of performing** (by the feedback provided by the polling of the CPU to the GPU the indication that the GPU is capable of performing the task or if the task needs to be partitioned is complete and disclosed in [0050] - *The information in the feedback array can be used by a graphics driver program (or another program executing on CPU 102) for load balancing, as illustrated in FIG. 5. Process 500 is a shown as a continuous loop in which the relative load on the GPUs is estimated from time to time by averaging values stored in the feedback array and the load is adjusted based on the estimate…*).

16. Regarding claim 21, it is similar in scope to claim 20 and is rejected under the same rationale.

17. Regarding claim 22, it is similar in scope to claim 20 and is rejected under the same rationale.

18. Regarding claim 23, it is similar in scope to the combination of claims 8, 11, 14, 17, and 20 (the rationale disclosed in the rejection incorporated herein). However, claim 23 includes the additional limitations of a method for creating a rendered polygon, **receiving a request to render a polygon; creating a representation of said rendered polygon comprising a root GPU program and its relationship with other GPU programs, their inputs and outputs; calling the following groups of objects for each GPU program that must be run in order that the root GPU program may run to render said polygon; one or more objects for creating a buffer**.

Boudier teaches **receiving a request to render polygon; creating a representation of said rendered polygon comprising a root node and its relationship with other nodes, their inputs and outputs; calling the following groups of objects for each node that must be run in order that the root node may run to render said polygon; one or more objects for creating a buffer** (the rendering of Boudier is created by creating a platform to optimize the processing and then executing the platform whereupon an image or objected is rendered, by the nature of rendering graphics is implied that polygons are rendered in order to create the make up of the object or image; [0005] - *The invention described herein is a system, method, and computer program product for optimization of a scene graph. The system of the invention includes an optimization base that contains a set of specific atomic*

optimizations… [0006] - *The system also includes an optimization registry that lists each atomic optimization, parameters associated with each optimization, and priority information relating to the necessary order in which optimizations must be performed. The system also includes an optimization manager which creates, configures, and applies an optimization process to an input scene graph. The system further includes an optimization configuration module for accepting user input to the optimization process…* [0007] - *The method of the invention includes the steps of receiving an input scene graph, creating the optimization process, applying the optimization process to the input scene graph, and post-optimization processing. The optimization process can be performed for any of a number of purposes, such as the enhancement of scene graph traversal time, the enhancement of drawing time, the reduction of memory usage, improved efficiency of data manipulation, and the targeting of a specific rendering platform…).*

Nash is analogous art that further teaches of **wherein the node(s) are program(s)** ([Fig. 3] and [0054]; furthermore [Fig. 6] shows phase modules (nodes) of processing and each node comprises a specific program code to conduct the function within said node in a sequence) **and wherein executing the modules yields a rendered image** ([0060], [0061] and [0062] - *The phase modules, containing microinstructions for a phase or sub-phase, are stored in phase code depository 122… The phase modules (excluding the output phase module) are interchangeable and capable of being synthesized in any order to implement a desired mode… Typically in*

computer graphics, the user supplies function calls to set the vertices, normals, primitives, textual coordinates, other operations, or the like to render an image…).

All the elements of claim 23 are known in Boudier in view of Nash, the only difference is the combination of known elements into a single system and method.

Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to include wherein the nodes of Boudier are modules indicating a function that needs to be processed and wherein the execution of the modules renders an image in Boudier, as doing so would provide the means for effectively optimizing the calculations of the attributes to a specific image instead of only optimizing the steps of applying the attributes, reducing overall system calculations and increases system efficiency and bandwidth.

Boudier and Nash teach the limitations of claim 8 above; however, Boudier and Nash fail to specifically teach of wherein the **programs can be run on a GPU** (GPU programs) and therefore the compilation is conducted **on a GPU.**

Diard is analogous art that further teaches of wherein the **programs can be run on a GPU** (GPU programs) and therefore the compilation is conducted **on a GPU** ([0033] - *In operation, a graphics driver program (or other program) executing on CPU 102 delivers rendering commands and associated data for processing by GPUs 114a, 114b… a rendering command may be associated with rendering data, with the rendering command defining a set of rendering operations to be performed by the GPU on the associated rendering data. In some embodiments, the rendering data is stored in the command buffer adjacent to the associated rendering command.*).

All the elements of claim 23 are known in Boudier and Nash in view of Diard, the only difference is the combination of known elements into a single system and method.

Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to include in Boudier the implementation of processing the compilation on a GPU as suggested by Diard, as doing so would provide the means and advantages of a higher throughput optimized, faster processing capabilities, and more advantageous memory bandwidth.

19. Regarding claim 24, the rationale disclosed in claim 23 is incorporated herein.

20. Regarding claim 25, Boudier, Nash, and Diard teach the limitations of claim 23 above, the rationale disclosed in the rejection incorporated herein and Boudier further teaches **wherein said representation of said rendered polygon is a low-level graph** (the low level graph is disclosed by fig. 9 (left side) wherein it has not been collapsed).

21. Regarding claim 26, Boudier, Nash, and Diard teach the limitations of claim 23 above, the rationale disclosed in the rejection incorporated herein and Boudier further teaches **wherein said representation of said rendered polygon is a high-level graph** (the high level graph is disclosed by fig. 9 (right side) wherein it has been collapsed).

22. Regarding claim 27, the rationale disclosed in the rejection of claim 23 is incorporated herein.

23. Regarding claim 28, the rationale disclosed in the rejection of claim 23 is incorporated herein.

24. Regarding claim 29, the rationale disclosed in the rejection of claim 23 is incorporated herein.

25. Regarding claim 30, the rationale disclosed in the rejection of claim 23 is incorporated herein.

26. Regarding claims 31 and 34, they are similar in scope to claim 23 and is rejected under the same rationale.

27. Regarding claim 32, it is similar in scope to claim 25 and is rejected under the same rationale.

28. Regarding claim 33, it is similar in scope to claim 26 and is rejected under the same rationale.

29. Regarding claim 35, it is similar in scope to claim 27 and is rejected under the same rationale.

30. Regarding claim 36, it is similar in scope to claim 28 and is rejected under the same rationale.

31. Regarding claim 37, it is similar in scope to claim 29 and is rejected under the same rationale.

32. Regarding claim 38, it is similar in scope to claim 30 and is rejected under the same rationale.

33. Regarding claim 39, it is similar in scope to claims 8, 20, 23 or 31 (the rationale disclosed in the rejection incorporated herein. However, claim 39 includes the additional limitation of **providing a computer-readable medium with executable instructions**. Boudier further teaches of **providing a computer-readable medium with executable instructions** [0038] - *Computer system 500 also includes a main memory 508, preferably random access memory (RAM), and may also include a secondary memory 510. The secondary memory 510 may include, for example, a hard disk drive 512 and/or a removable storage drive 514, representing a magnetic tape drive, an optical disk drive, etc. The removable storage drive 514 reads from and/or writes to a removable storage unit 518 in a well known manner. Removable storage unit 518 represents a magnetic tape, optical disk, etc. As will be appreciated, the removable storage unit 518 includes a computer usable storage medium having stored therein computer software and/or data.*).

Response to Arguments

[Objection to the Specification]

34. Applicant's arguments, see Remarks (part a, page 10), filed 1/06/2009, with respect to the objection to the specification regarding "computer readable medium" have

been fully considered and are persuasive. The objection of the specification in this regard has been withdrawn.

[Claim rejections - Section 103(a)]

35. Applicant's arguments filed 1/06/2009 have been fully considered but they are not persuasive.

Independent claims 8, 23, 31, and 39

Applicant's arguments with respect to claims 8, 23, 31, and 39 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to AARON M. GUERTIN whose telephone number is (571)270-1547. The examiner can normally be reached on M-F 8:30AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Xiao Wu can be reached on 571-272-7761. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Aaron M Guertin/
Examiner, Art Unit 2628

/XIAO M. WU/
Supervisory Patent Examiner, Art Unit 2628